Experiment No.6
Implement various join operations
Date of Performance:
Date of Submission:

Aim :- Write simple query to implement join operations(equi join, natural join, inner join, outer joins).

Objective :- To apply different types of join to retrieve queries from the database management system.

Theory:

SQL Join statement is used to combine data or rows from two or more tables based on a common field between them. Different types of Joins are as follows:

- INNER JOIN
- LEFT JOIN
- RIGHT JOIN
- FULL JOIN

A. INNER JOIN

The INNER JOIN keyword selects all rows from both the tables as long as the condition is satisfied. This keyword will create the result-set by combining all rows from both the tables where the condition satisfies i.e value of the common field will be the same.

Syntax:

SELECT table1.column1,table1.column2,table2.column1,....

FROM table1

INNER JOIN

table2

ON table1.matching_column = table2.matching_column;

table1: First table.

table2: Second table

matching_column: Column common to both the tables.

B. LEFT JOIN

This join returns all the rows of the table on the left side of the join and matches rows for the table on the right side of the join. For the rows for which there is no matching row on the right side, the result-set will contain *null*. LEFT JOIN is also known as LEFT OUTER JOIN.

Syntax:

SELECT table1.column1,table1.column2,table2.column1,....

FROM table1

LEFT JOIN table2

ON table1.matching_column = table2.matching_column;

table1: First table.

table2: Second table

matching_column: Column common to both the tables.

C. RIGHT JOIN

RIGHT JOIN is similar to LEFT JOIN. This join returns all the rows of the table on the right side of the join and matching rows for the table on the left side of the join. For the rows for which there is no matching row on the left side, the result-set will contain *null*. RIGHT JOIN is also known as RIGHT OUTER JOIN.

Syntax:

SELECT table1.column1,table1.column2,table2.column1,....

FROM table1

RIGHT JOIN

table2

ON table1.matching_column = table2.matching_column;

table1: First table.

table2: Second table

matching_column: Column common to both the tables.

D. FULL JOIN

FULL JOIN creates the result-set by combining results of both LEFT JOIN and RIGHT JOIN. The result-set will contain all the rows from both tables. For the rows for which there is no matching, the result-set will contain NULL values.

Syntax:

SELECT table1.column1,table1.column2,table2.column1,....

FROM table1

FULL JOIN table2

ON table1.matching_column = table2.matching_column;

table1: First table.

table2: Second table

matching_column: Column common to both the tables.

Implementation:

inner join

customer_id	customer_name	city	pincode	house_no	customer_mob	customer_email	bill_id	bill_customer_id	bill_desc	bill_type	bill_num
98765	Saurabh	Vasai	401205	210	456232232	saurabh@gmail.com	847674	98765	mobile	ordinary bill	2
1234567	Harsh	Mumbai	401202	201	123456789	harrsh@gmail.com	5134513	1234567	clothes	ordinary bill	1

customer_id	customer_name	city	pincode	house_no	customer_mob	customer_email	bill_id	bill_customer_id	bill_desc	bill_type	bill_num
98765	Saurabh	Vasai	401205	210	456232232	saurabh@gmail.com	847674	98765	mobile	ordinary bill	2
1234567	Harsh	Mumbai	401202	201	123456789	harrsh@gmail.	harrsh@gmail.com	34567	clothes	ordinary bill	1

left join

customer_id	customer_name	city	pincode	house_no	customer_mob	customer_email	bill_id	bill_customer_id	bill_desc	bill_type	bill_num
98765	Saurabh	Vasai	401205	210	456232232	saurabh@gmail.com	847674	98765	mobile	ordinary bill	2
1234567	Harsh	Mumbai	401202	201	123456789	harrsh@gmail.com	5134513	1234567	clothes	ordinary bill	1

right join

customer_id	customer_name	city	pincode	house_no	customer_mob	customer_email	bill_id	bill_customer_id	bill_desc	bill_type	bill_num
98765	Saurabh	Vasai	401205	210	456232232	saurabh@gmail.com	847674	98765	mobile	ordinary bill	2
1234567	Harsh	Mumbai	401202	201	123456789	harrsh@gmail.com	5134513	1234567	clothes	ordinary bill	1

full join

Conclusion:

1. Illustrate how to perform natural join for the joining attributes with different names with a suitable example.
2. Illustrate significant differences between natural join equi join and inner join.

In conclusion, when joining tables with different column names using a natural join, the database system automatically matches columns with the same name in both tables. For example: -- Consider two tables: employees and departments -- employees table has columns: employee_id, employee_name, department_id -- departments table has columns: dept_id, dept_name -- Performing a natural join between employees and departments SELECT * FROM employees NATURAL JOIN departments; In this scenario, the natural join operation matches the "department_id" column from the employees table with the "dept_id" column from the departments table, even though they have different names. Additionally, significant differences exist between natural join, equi join, and inner join: 1. Natural Join: - Automatically performs the join based on columns with the same name in both tables. - Does not require explicit specification of join conditions. - May lead to unexpected results if there are columns with the same name but different meanings in the tables. 2. Equi Join: - Explicitly specifies the join condition using the equality operator (=). - Can join tables based on columns with different names or using complex conditions. - Allows for greater control over the join operation and ensures clarity in the query. 3. Inner Join: - Retrieves only the rows from both tables that satisfy the join condition. - Can be performed using either the ON clause or the WHERE clause. - Does not include rows from either table that do not meet the join condition. In summary, while natural join automatically matches columns with the same name, equi join requires explicit specification of join conditions using the equality operator. Inner join retrieves only the rows that satisfy the join condition, regardless of the join type. Each type of join serves different purposes and provides different levels of control over the join operation